IMAGE PROCESSING APPARATUS SUPPORTING BOTH DISCRETE COSINE TRANSFORM AND DISCRETE WAVELET TRANSFORM

Background of the Invention

1. Field of the Invention

The present invention is related to an apparatus and method for image processing, in particular, to an image processor adaptive to a plurality of coding and decoding procedures.

2. Description of the Related Art

The JPEG (Joint Photographic Expert Group) algorithm, which uses a discrete cosine transform (DCT), is one of the most common static image compressing methods. The JPEG algorithm achieves high compression with reduced image deterioration, and thereby allows personal computers and facsimiles to process image data with reduced hardware resources.

The JPEG algorithm, however, suffers from several drawbacks, including image deterioration at low bit rates.

In order to overcome these drawbacks, JPEG 2000 algorithm has been recently standardized and become commercially available. The JPEG 2000 algorithm employs a discrete wavelet transform

(DWT) to code and decode image data in place of the discrete cosine transform.

This situation requires image processing apparatuses to support both of the conventional JPEG and JPEG 2000 algorithms. Japanese Unexamined Patent Application No. 2001-103484 discloses an image processing apparatus selectively performing the DCT and DWT to be adaptive to the conventional JPEG and JPEG 2000 algorithms. Fig. 1 shows a block diagram of the disclosed image processing apparatus. The disclosed image processing apparatus is composed of an input selector 50, a DCT processor 51, a DWT processor 52, and an output selector 53. The input selector 50 selects one of the DCT processor 51, the DWT processor 52 in response to a selection signal received from a circuit, and transfers input data to the selected processor. The DCT processor 51 encodes the data received from the input selector 50 using the discrete cosine transform, while the DWT processor 52 encodes the data received from the input selector 50 using the discrete wavelet transform. The output selector 53 outputs the encoded data in response to the selector signal.

Japanese Unexamined Patent Application No. H06-46404 discloses an image data processing

apparatus for reducing image derogation in image edges. This image data processing apparatus detects image edges in units of image blocks, and encodes the image block(s) including the detected image edge(s) using the wavelet transform in place of the discrete cosine transform.

An issue of the conventional image processing apparatuses is that they requires large hardware resources to support both the DCT and DWT algorithms.

A need exists to provide an image processing apparatus which supports both the DCT and DWT algorithms with reduced hardware resources.

Summary of the Invention

Therefore, an object of the present invention is to provide an image processing apparatus which supports both the DCT and DWT algorithms with reduced hardware resources.

In an aspect of the present invention, an image processing apparatus is composed of an input unit receiving a plurality of pixel data, a controlling unit selecting a desired transform from among discrete wavelet transform and discrete cosine transform, and providing a plurality of coefficients depending on the

desired transform, and a processing unit which processes the pixel data using the plurality of coefficients to achieve the desired transform.

The input unit preferably includes a storage unit storing the pixel data, and a rearrangement unit receiving and rearranging the pixel data so as to be adaptive to the desired transform in response to a control signal received from the control unit. The processing unit processes the rearranged pixel data to achieve the desired transform.

The processing unit preferably includes a plurality of adders, a plurality of multipliers, and an adder/subtractor unit. Each of the plurality of adders calculates a sum of two of the rearranged pixel data, the two of the rearranged pixel data being selected by the rearranged unit. Each of the plurality of multipliers calculates a product of associated one of the sums and associated one of the plurality of the coefficients. The adder/subtractor unit executes operation on the products received from the plurality of multipliers to obtain a result data of the desired transform.

It is advantageous if the controlling unit selects one procedure from among encoding and

decoding through the desired transform, and develops the plurality of coefficients depending on the selected procedure.

It is also advantageous if the controlling unit selects one procedure from among encoding and decoding through the desired transform, and develops the control signal to allow the rearrangement unit to be adaptive to the selected procedure.

Preferably, the controlling unit selects one of an irreversible 9/7 filter and a reversible 5/3 filter to be used when selecting the discrete wavelet transform, and develops the plurality of coefficients depending on the selected filter.

It is also preferable that the controlling unit selects one of an irreversible 9/7 filter and a reversible 5/3 filter to be used when selecting the discrete wavelet transform, and develops the control signal to allow the rearrangement unit to be adaptive to the selected procedure.

The input unit may include a plurality of flipflops which respectively stores therein one of the plurality of pixel data, a rearrangement unit receiving the plurality of pixel data from the plurality of flipflops and rearranging the

received pixel data so as to be adaptive to the desired transform in response to a control signal received from the control unit, and the processing unit may includes a plurality of adders, each receiving two of the plurality of pixel data selected by the rearrangement unit to calculate a sum of the received two pixel data, a plurality of multipliers, each calculating a product of associated one of the sums and associated one of the plurality of the coefficients, another multiplier receiving one of the plurality of pixel data from one of the flipflops and calculating a product of the received pixel data and associated one of the plurality of the coefficients, a selector; and an adder/subtractor unit, the selector selecting one of outputs of the another multiplier and the adder/subtractor unit, and the adder/subtractor unit executing operation on the products received from the plurality of multipliers and an output of the selector to obtain a result data of the desired transform.

In another aspect of the present invention, an image processing method is composed of:

receiving a plurality of pixel data;

selecting a desired transform from among discrete wavelet transform and discrete cosine

transform;

providing a plurality of coefficients depending on the desired transform; and

processing the pixel data using the set of coefficients to achieve the desired transform.

The image processing method preferable further includes:

rearranging the pixel data so as to be adaptive to the desired transform, wherein the processing is executed with respect to the rearranged pixel data to achieve the desired transform.

the processing preferably includes:

providing pixel data pairs each including two of the rearranged pixel data,

calculating sums of respective pixel data pairs,

calculating products of the sums and the plurality of coefficients;

executing operation on the products to obtain a result data of the desired transform.

The image processing method preferably includes:

selecting one procedure from among encoding and decoding through the desired transform, wherein the plurality of coefficients are developed depending on the selected procedure.

The image processing method preferably includes:

selecting one procedure from among encoding and decoding through the desired transform, the rearranging the pixel data being executed depending on the selected desired procedure.

The image processing method preferably includes:

selecting one of an irreversible 9/7 filter and a reversible 5/3 filter to be used when selecting the discrete wavelet transform, the plurality of coefficients being developed depending on the selected filter.

The image processing method preferably includes:

selecting one of an irreversible 9/7 filter and a reversible 5/3 filter to be used when selecting the discrete wavelet transform, the rearranging being executed depending on the selected procedure.

Brief Description of the Drawings

Fig. 1 is a block diagram of a conventional image processing apparatus;

Fig. 2 is a block diagram of an image processing apparatus in a first embodiment of the present invention;

Fig. 3 is a detailed block diagram of the image processing apparatus in the first embodiment;

Fig. 4 is a table illustrating a set of coefficients provided for the multiplier unit 23 from the controller unit 30;

Figs. 5 to 7 are timing diagrams illustrating encoding through the discrete cosine transform;

Figs. 8 to 10 are timing diagrams illustrating decoding through the discrete cosine transform;

Fig. 11 is a block diagram of an image processing apparatus in a second embodiment;

Fig. 12 is a timing diagram illustrating encoding through discrete wavelet transform using a reversible 5/3 filter in the second embodiment;

Fig. 13 is a timing diagram illustrating decoding through discrete wavelet transform using a reversible 5/3 filter in the second embodiment; and

Fig. 14 is a block diagram illustrating a reversible circuit within the image processing apparatus in the second embodiment.

D scription of the Pr ferred Embodiments

Preferred embodiments of the present

invention are described below in detail with reference to the attached drawings.

DWT and DCT algorithms

An image processing apparatus in accordance with the present invention encodes and decodes image data using discrete wavelet transform and discrete cosine transform. Below is an explanation of the discrete wavelet transform and the discrete cosine transform used in this embodiment.

The image processing apparatus is adapted to discrete wavelet transform using an irreversible 9/7 filter and/or a reversible 5/3 filter described in the following.

The DCT algorithm using the irreversible 9/7 filter is characterized in that the coefficients of the filter are real numbers and that the DWT algorithm fails to perform rounding of the encoded and decoded image data.

The DCT algorithm using the irreversible 9/7 filter encodes pixel data of even numbered columns of pixels in the image in accordance with the following equation(1):

$$Y(2n)=W1*(X(2n-4)+X(2n+4))-W0*(X(2n-3)+X(2n+3))$$
$$-W3*(X(2n-2)+X(2n+2))+W5*(X(2n-1)+X(2n+1))$$
$$+W7*X(2n), \quad \cdots(1)$$

where X(i) is an original pixel data, and Y(i) is an encoded pixel data, while encoding pixel data of odd numbered columns in accordance with the following equation (2):

$$Y(2n+1)=W4*(X(2n-2)+X(2n+4))-W2*(X(2n-1)+X(2n+3))$$
$$-W6*(X(2n)+X(2n+2))+W8*X(2n+1), \quad \cdots(2)$$

where W0 through W7 are filter coefficients of the irreversible 9/7 filter given in the following:

W0 = 0.0168641184…,

W1 = 0.0267487574…,

W2 = 0.0575435262…,

W3 = 0.0782232665…,

W4 = 0.0912717631…,

W5 = 0.2668641184…,

W6 = 0.5912717631…,

W7 = 0.6029490182…, and

W8 = 1.1150870524…

The DWT algorithm using the irreversible 9/7 filter, on the other hand, decodes pixel data of the even numbered columns of the pixels in accordance with the following equation(3):

$$X(2n)=W0*(Y(2n-3)+Y(2n+3))-W2*(Y(2n-2)+Y(2n+2))$$
$$-W5*(Y(2n-1)+Y(2n+1))+W8*Y(2n), \quad \cdots(3)$$

while decoding pixel data of the odd numbered columns in accordance with the following equation (4):

$$X(2n+1)=W1*(Y(2n-3)+Y(2n+5))-W4*(Y(2n-2)+Y(2n+4))$$
$$-W3*(Y(2n-1)+Y(2n+3))+W6*(Y(2n)+Y(2n+2))$$
$$+W7*Y(2n+1), \quad \cdots(4)$$

where W0 through W7 are the above-described filter coefficients.

The DWT algorithm using the reversible 5/3 filter, on the other hand, is characterized in that the coefficients of the filter are integers and that the DWT algorithm performs rounding of the encoded and decoded image data to integerize.

The DWT algorithm using the reversible 5/3 filter encodes pixel data of even numbered columns of pixels in the image in accordance with the following equation (5):

$$Y(2n+1)=X(2n+1)-\left[\frac{X(2n)+X(2n+2)}{2}\right], \quad \cdots(5)$$

while encoding pixel data of odd numbered columns of pixels in accordance with the following equation (6):

$$Y(2n)=X(2n)-\left[\frac{Y(2n-1)+Y(2n+1)+2}{4}\right], \quad \cdots(6)$$

where [x] is the floor function defined as follows: for a real number x, [x] is the largest integer less than or equal to x,

The DWT algorithm using the reversible 5/3

filter, on the other hand, decodes pixel data of the even numbered columns of pixels in accordance with the following equation (7):

$$X(2n)=Y(2n)-\left[\frac{Y(2n-1)+Y(2n+1)+2}{4}\right], \quad \cdots(7)$$

while decoding pixel data of the odd numbered columns of pixels in accordance with the following equation (8):

$$X(2n+1)=Y(2n+1)-\left[\frac{X(2n)+X(2n+2)}{2}\right] \quad \cdots(8)$$

The image processing apparatus also performs a DCT algorithm described in the following. The DCT algorithm encodes pixel data of even numbered columns of pixels in accordance with the following equation (9):

$$\begin{bmatrix} F0 \\ F4 \\ F2 \\ F6 \end{bmatrix} = \begin{bmatrix} a0+a1+a3+a2 & 0 & 0 \\ a0-a1+a3-a2 & 0 & 0 \\ 0 & a0-a3 & a1-a2 \\ 0 & -a1+a2 & a0-a3 \end{bmatrix} \begin{bmatrix} D3 \\ D5 \\ D1 \end{bmatrix}, \quad \cdots(9)$$

while encoding pixel data of odd numbered columns of pixels in accordance with the following equation (10):

$$\begin{bmatrix} F1 \\ F3 \\ F5 \\ F7 \end{bmatrix} = \begin{bmatrix} a4 & a5 & a6 & a7 \\ -a6 & a4 & -a7 & -a5 \\ -a5 & a7 & a4 & a6 \\ -a7 & a6 & -a5 & a4 \end{bmatrix} \begin{bmatrix} D6 \\ D4 \\ D2 \\ D0 \end{bmatrix}, \quad \cdots(10)$$

where F0 through F7 are encoded pixel data, D0 through D6 are filter coefficients defined as follows:

D0 = 0.19509032,

D1 = 0.38268343,

D2 = 0.55557023,

D3 = 0.70710678,

D4 = 0.83146961,

D5 = 0.92387953, and

D6 = 0.98078528,

and a0 through a7 are coefficients defined as follows:

$$\begin{aligned} a0 &= f0 + f7, \\ a1 &= f1 + f6, \\ a2 &= f2 + f5, \\ a3 &= f3 + f4, \\ a4 &= f0 - f7, \\ a5 &= f1 - f6, \\ a6 &= f2 - f5, \text{ and} \\ a7 &= f3 - f4, \end{aligned} \qquad \cdots(11)$$

where f0 through f7 are original pixel data.

The DCT algorithm decodes pixel data of the even numbered columns of pixels in accordance with the following equations (12) and (13):

$$\frac{1}{2}\begin{bmatrix} f0+f4 \\ f1+f5 \\ f2+f6 \\ f3+f7 \end{bmatrix} = \begin{bmatrix} a0+a3 & a2 & a1 \\ a0-a3 & -a1 & a2 \\ a0-a3 & a1 & -a2 \\ a0+a3 & -a2 & -a1 \end{bmatrix} \begin{bmatrix} D3 \\ D5 \\ D1 \end{bmatrix}, \quad \text{and} \quad \cdots(12)$$

$$\frac{1}{2}\begin{bmatrix} f0-f4 \\ f1-f5 \\ f2-f6 \\ f3-f7 \end{bmatrix} = \begin{bmatrix} a5 & a7 & -a6 & -a4 \\ a6 & a5 & a4 & -a7 \\ -a7 & -a4 & a5 & -a6 \\ a4 & -a6 & -a7 & a5 \end{bmatrix} \begin{bmatrix} D6 \\ D4 \\ D2 \\ D0 \end{bmatrix}, \qquad \cdots(13)$$

where a0 through a7 are coefficients defined as follows:

$$
\begin{aligned}
&a0 = F0,\\
&a1 = F6,\\
&a2 = F2,\\
&a3 = F4, \qquad \cdots(14)\\
&a4 = -F7,\\
&a5 = F1,\\
&a6 = -F5, \quad \textit{and}\\
&a7 = F3.
\end{aligned}
$$

First Embodiment

Fig. 1 shows a block diagram of an image processing apparatus in a first embodiment. The image processing apparatus in this embodiment is designed to support both discrete cosine transform and discrete wavelet transform using an irreversible 9/7 filter.

The image processing unit in this embodiment, is composed of an input unit 10, a processing unit 20, and a controller unit 30, which are monolithically integrated within an LSI (large scale integrated circuit).

The input unit 10 includes a storage unit 11 and a rearranging circuit 12. The storage unit 11 stores therein pixel data received from an external device. The storage unit 11 transfers the stored pixel data to the rearranging circuit 12. As described below, the storage unit 11 is composed of a shift register.

The rearranging circuit 12 rearranges the

order of the pixel data received from the storage unit 11 so that the order of the pixel data is adaptive to the discrete wavelet transform or the discrete cosine transform in response to a control signal received from the controller unit 30. The rearranging circuit 12 defines pixel data pairs, which are different two of the pixel data. The rearranged pixel data is transferred to the processing unit 20.

The processing unit 20 is composed of an adder unit 21, a multiplier unit 22, and an adder/subtractor unit 23. The adder unit 21 calculates sums of the respective pixel data pairs or differences between the respective pixel data pairs, in response to a control signal S2 received from the controller unit 30. For the discrete wavelet transform in accordance with the equation (1), for example, the adder unit 21 calculates the sums a1 to a4 in parallel as follows:

$$a1 = X(2n-4)+X(2n+4),$$

$$a2 = X(2n-3)+X(2n+3),$$

$$a3 = X(2n-2)+X(2n+2), \text{ and}$$

$$a4 = X(2n-1)+X(2n+1).$$

The calculated sums are transferred to the multiplier unit 23.

The multiplier unit 23 receives a control

signal S2 representative of filter coefficients from the controller unit 30, and calculates respective products of the sums received from the adder unit 21 and the associated filter coefficients received from the controller unit 30. For the discrete wavelet transform in accordance with the equation (1), for example, the multiplier unit 23 calculates the products MPY1 to MPY5 in parallel as follows:

MPY1 = W1 x a1,

MPY2 = W0 x a2,

MPY3 = W3 x a3,

MPY4 = W5 x a4, and

MPY5 = W7 x X(2n).

The calculated products are transferred to the adder/subtractor unit 25.

The adder/subtractor unit 25 is responsive to a control signal S3 received from the controller unit 30 for calculating addition and/or subtraction with respect to the products MPY1 to MPY5, and thereby obtains the encoded or decoded pixel data. For the discrete wavelet transform in accordance with the equation (1), for example, the adder/subtractor unit 25 calculates the encoded pixel data Y(2n) defined by the following equation:

Y(2n) = MPY1-MPY2-MPY3+MPY4+MPY5.

The same goes for the equations (2) to (4) and the equations (9), (10), (12) and (13).

The controller unit 30 provides the control signals S1 for the input unit 10 and the control signals S2 to S4 for the processing unit 20. The controller unit 30 determines which operation is to be performed, and indicate the input unit 10 and the processing unit 20 to perform the determined operation by providing the control signals S1 to S4. The determined operation includes: encoding and decoding through the discrete wavelet transform using the irreversible 9/7 filter, and encoding and decoding through the discrete cosine transform.

Fig. 2 shows a detailed block diagram of the processing apparatus in this embodiment. The processing unit 20 includes latches 22 and 24, a flipflop FF19, a limiter 26, and a flipflop 20 in addition to the adder unit 21, the multiplier unit 23, and the adder/subtractor unit 25.

The storage device 11 is composed of a flipflop FF_{inp} and a set of flipflops FF0 through FF8. The flipflops FF0 to FF8 are provided to store pixel data f0 to f8. The pixel data f4 is associated with the pixel of interest of the discrete wavelet transform, and the pixel data f0 to f3, and f5 to f8 are associated with the

pixels adjacent to the pixel of interest.

The flipflop FF_{inp} functions as a buffer receiving the pixel data to be encoded or decoded. The output of the flipflop FF_{inp} is connected to the input of the flipflop FF0. The flipflops FF0 through FF8 are connected in serial to constitute a shift register. The flipflop FF_{inp} and flipflops FF0 through FF8 receives the same clock signal (not shown) and operates in synchronization with the clock signal. The received pixel data is transferred through the flipflops FF0 and FF8 in response to the clock signal. In an alternative embodiment, the flipflops FF0 and FF8 may directly receive the pixel data in parallel. The outputs of the respective flipflops FF0 to FF8 are connected to the rearrangement circuit 12.

The rearrangement circuit 12 rearranges the order of the pixel data f0 to f8 as indicated by the control signal S1 from the controller unit 30 to provide a set of pixel data m1 to m8. The rearrangement of the pixel data f0 to f8 depends on which transform is to be performed.

The row indicated by the symbol "9/7 encoding" represents which pixel data are outputted as the respective pixel data m1 to m8 for the encoding through the discrete wavelet transform using the irreversible 9/7 filter. The

rearrangement of the pixel data is executed depending on whether the pixel of interest is positioned in the even numbered columns or in the odd numbered columns. In detail, the rearrangement circuit 12 outputs the pixel data f8 as the pixel data m1, regardless of the position of the pixel of interest. The rearrangement circuit 12 outputs the pixel data f0 as the pixel data m2 in the event that the pixel of interest is positioned in the even numbered column; otherwise the rearrangement circuit 12 outputs zero as the pixel data m2. The rearrangement circuit 12 outputs the pixel data f7, f1, f6, f2, f5 and f3 as the pixel data m3, m4, m5, m6, m7, and m8, respectively, regardless of the position of the pixel of interest.

Correspondingly, the row indicated by the symbol "9/7 decoding" represents which pixel data are outputted for the decoding through the discrete wavelet transform using the irreversible 9/7 filter. The rearrangement circuit 12 outputs zero as the pixel data m1 in the event that the pixel of interest is positioned in the even numbered column; otherwise, the rearrangement circuit 12 outputs the pixel data f8 as the pixel data m1. The rearrangement circuit 12 outputs zero as the pixel data m2 in the event that the

pixel of interest is positioned in the even numbered column; otherwise the rearrangement circuit 12 outputs the pixel data f0 as the pixel data m2. And, the rearrangement circuit 12 outputs the pixel data f7, f1, f6, f2, f5 and f3 as the pixel data m3, m4, m5, m6, m7, and m8, respectively, regardless of the position of the pixel of interest.

The adder unit 21 is composed of a set of adders 21_1 to 21_4. The adder 21_1 calculates the sum of the pixel data m1 and m2. The sum of the pixel data m1 and m2 is denoted by the numeral "a1" or "a5". Correspondingly, the adder 21_2, 21_3, and 21_4 calculate the sum of the pixel data m3 and m4, the sum of the pixel data m5 and m6, and the sum of the pixel data m7 and m8, respectively. The sum of the pixel data m3 and m4 is denoted by the numeral "a2" or "a6", the sum of the pixel data m5 and m6 is denoted by the numeral "a3" or "a7", and the sum of the pixel data m7 and m8 is denoted by the numeral "a4" or "a8". The calculated sums a1 through a8 are transferred to the latch 22.

The latch 22 is composed of a set of flipflops FF9 and FF13. The flipflop FF9 latches the sum a1 (or a5) received from the adder 21_1 and transfers the latched sum a1 (or a5) to the

multiplier unit 23. The flipflop FF10 latches the sum a2 (or a6) received from the adder 21_2, and transfers the latched sum a2 (or a6) to the multiplier unit 23. The flipflop FF11 latches the sum a3 (or a7) received from the adder 21_3, and transfers the latched sum a3 (or a7) to the multiplier unit 23. The flipflop FF12 latches the sum a4 (or a8) received from the adder 21_4, and transfers the latched sum a4 (or a8) to the multiplier unit 23. The flipflop FF13 latches the pixel data f4 from the flipflop FF4, and transfers the latched pixel data to the multiplier unit 23.

The multiplier 23 is composed of a set of multipliers 23_1 to 23_5. The multiplier 23_1 calculate a product MPY1 of the sum a1 (or a5) received from the flipflop FF9 and a coefficient α described in the control signal S3 from the controller unit 30. The product MPY1 is transferred to the latch 24. Correspondingly, the multiplier 23_2 calculate a product MPY2 of the sum a2 (or a6) received from the flipflop FF10 and a coefficient β described in the control signal S3 from the controller unit 30. The product MPY2 is transferred to the latch 24. The multiplier 23_3 calculate a product MPY3 of the sum a3 (or a7) received from the flipflop FF11 and a coefficient

γ described in the control signal S3 from the controller unit 30. The product MPY3 is transferred to the latch 24. The multiplier 23_4 calculate a product MPY3 of the sum a4 (or a8) received from the flipflop FF12 and a coefficient δ described in the control signal S3 from the controller unit 30. The product MPY4 is transferred to the latch 24. And the multiplier 23_5 calculate a product MPY5 of the pixel data f4 received from the flipflop FF13 and a coefficient ε described in the control signal S3 from the controller unit 30. The product MPY5 is transferred to the latch 24.

As shown in Fig. 3, the coefficient α depends on the kind of the transfer to be performed as described in the following. For encoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient α is set to the aforementioned coefficient W1 in the event that the pixel of interest is positioned in the even numbered columns, while set to zero (0) in the event that the pixel of interest is positioned in the odd numbered columns. For decoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient α is set to zero in the event that the pixel of interest is

positioned in the even numbered columns, while set to W1 in the event that the pixel of interest is positioned in the odd numbered columns. For both encoding and decoding through the discrete cosine transform, the coefficient α is set to D0 in the event that the pixel of interest is positioned in the even numbered columns, while set to D5 in the event that the pixel of interest is positioned in the odd numbered columns.

Correspondingly, the coefficients β through ε depend on the kind of the transfer to be performed as described in the following. For encoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient β is set to -W0 in the event that the pixel of interest is positioned in the even numbered columns, while set to W4 in the event that the pixel of interest is positioned in the odd numbered columns. For decoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient β is set to W0 in the event that the pixel of interest is positioned in the even numbered columns, while set to -W4 in the event that the pixel of interest is positioned in the odd numbered columns. For both encoding and decoding through the discrete cosine transform, the coefficient β is set to D1 in the

event that the pixel of interest is positioned in the even numbered columns, while set to D4 in the event that the pixel of interest is positioned in the odd numbered columns.

For encoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient γ is set to -W3 in the event that the pixel of interest is positioned in the even numbered columns, while set to -W2 in the event that the pixel of interest is positioned in the odd numbered columns. For decoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient γ is set to -W3 in the event that the pixel of interest is positioned in the even numbered columns, while set to -W2 in the event that the pixel of interest is positioned in the odd numbered columns. For both encoding and decoding through the discrete cosine transform, the coefficient γ is set to D3 in the event that the pixel of interest is positioned in the even numbered columns, while set to D2 in the event that the pixel of interest is positioned in the odd numbered columns.

For encoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient δ is set to W5 in the event that the

pixel of interest is positioned in the even numbered columns, while set to -W6 in the event that the pixel of interest is positioned in the odd numbered columns. For decoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient δ is set to -W5 in the event that the pixel of interest is positioned in the even numbered columns, while set to W6 in the event that the pixel of interest is positioned in the odd numbered columns. For both encoding and decoding through the discrete cosine transform, the coefficient δ is set to zero in the event that the pixel of interest is positioned in the even numbered columns, while set to D0 in the event that the pixel of interest is positioned in the odd numbered columns.

For encoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient ε is set to W7 in the event that the pixel of interest is positioned in the even numbered columns, while set to W8 in the event that the pixel of interest is positioned in the odd numbered columns. For decoding through the discrete wavelet transform using the irreversible 9/7 filter, the coefficient ε is set to W8 in the event that the pixel of interest is positioned in the even numbered columns, while set to W7 in the

event that the pixel of interest is positioned in the odd numbered columns. For both encoding and decoding through the discrete cosine transform, the coefficient ε is set to zero regardless of the position of the pixel of interest.

The latch 24 is composed of a set of flipflops FF14 through FF18. The flipflop FF14 latches the product MPY1 from the multiplier 24_1, and transfers the latched product MPY1 to the adder/subtractor unit 25. The flipflop FF15 latches the product MPY2 from the multiplier 24_2, and transfers the latched product MPY2 to the adder/subtractor unit 25. The flipflop FF16 latches the product MPY3 from the multiplier 24_3, and transfers the latched product MPY3 to the adder/subtractor unit 25. The flipflop FF17 latches the product MPY4 from the multiplier 24_4, and transfers the latched product MPY4 to the adder/subtractor unit 25. The flipflop FF18 latches the product MPY5 from the multiplier 24_5, and transfers the latched product MPY5 to the adder/subtractor unit 25.

The adder/subtractor unit 25 is composed of adder 25_1 through 25_4 and a selector 25_5. The selector 25_5 selects one of the data received from flipflops F8 and F19 in response to the control signal S4 received from the controller unit 30,

and outputs the selected data. The data from the flipflop F8 is selected for the discrete wavelet transform, while the data from the flipflop F19 is selected for the discrete cosine transform.

The adder 25_1 calculates the sum $\Sigma 1$ of the products MPY1 and MPY2 received from the flipflops FF14 and FF15, respectively. The adder 25_1 also calculates the sum $\Sigma 2$ of the products MPY2 and MPY3 received from the flipflops FF15 and FF16, respectively.

The adder 25_2 calculates the sum $\Sigma 3$ of the sum $\Sigma 2$ received from the adder 25_1 and the product MPY4 received from the flipflop FF17. The adder 25_2 also calculates the sum $\Sigma 4$ of the product MPY4 and the data received from the selector 25_5. The sums $\Sigma 3$ and $\Sigma 4$ are transferred to the adder 25_3.

The adder 25_3 calculates the sum $\Sigma 5$ of the sums $\Sigma 1$ and $\Sigma 3$ received from the adder 25_1 and the adder 25_2, respectively. The adder 25_3 also calculates the sum $\Sigma 6$ of the sums $\Sigma 3$ and $\Sigma 4$received from the adder 25_2.

The adder 25_4 calculates the sum $\Sigma 7$ of the sums $\Sigma 5$ and $\Sigma 6$ received from the adder 25_3. The sum $\Sigma 7$ is transferred to the flipflop FF19.

The flipflop FF19 latches the sum $\Sigma 7$ and transfers the latched sum $\Sigma 7$ to the selector 25_5

and the limiter 26.

The limiter 26 receives the sum $\Sigma 7$ from the flipflop FF19, and outputs an output data defined as follows: the output data is equal to the sum $\Sigma 7$ in the event that the sum $\Sigma 7$ is smaller than a specified value, while the output data is equal to the specified value in the event that the sum $\Sigma 7$ is equal to or larger than the specified value.

The flipflop FF20 latches the output data received from the limiter 26, and develops the latched output data on the output.

The aforementioned latches 22, 23, flipflops 19 and 20 allows the image processing apparatus to achieve pipeline processing. One skilled in the art would appreciate that the latches 22, 23, flipflops 19 and 20 may be removed in an alternative embodiment.

Below is an explanation of the operation of the image processing apparatus in this embodiment.

1-1) Procedure of encoding pixel data through the discrete wavelet transform using the irreversible 9/7 filter

This procedure begins with providing pixel data for the storage unit 11. It should be noted that "mirror" pixel data of "virtual pixels" may be provided for the storage unit 11 when the

pixel of interest is close to the end of the image The virtual pixels are defined as being pixels virtually disposed around the image, which are symmetrical to the pixels near the end of the image. The "mirror" pixel data are defined as the pixel data associated with the "virtual pixels". The pixel data associated with the pixel of interest is set to the flipflop FF4 of the storage unit 11.

After the pixel data f0 through f8 are respectively latched into the flipflop FF0 through FF8, the controller unit 30 develops the control signal S1 to indicate the rearrangement circuit 12 to perform the rearrangement of the pixel data f0 through f8 so that the order of the pixel data f0 through f8 are adapted to encoding through the discrete wavelet transform using the irreversible 9/7 filter. In response to the control signal S1, the rearrangement circuit 12 executes the rearrangement as indicated by the row denoted by "9/7 (ENCODING)". In the event that the pixel of interest is positioned in the even numbered columns, for example, the rearrangement circuit 12 outputs the pixel data f8, which is associated with X(2n+4) in the equation (1), as the pixel data m1, while outputting the pixel data f0, which is associated

with X(2n-4) in the equation (1), as the pixel data m2. The same goes for the pixel data m3 through m8. In the event that the pixel of interest is positioned in the odd numbered columns, the rearrangement circuit 12 outputs zero in place of the pixel data f0, as the pixel data m2.

The controller unit 30 then develops the control signal S2 to indicate the adder 21_1 to 21_4 to execute addition. The adder 21_1 calculates the sum a1 of the pixel data m1 and m2. The calculation of the sum a1 is equivalent to the calculation of the term "X(2n-4) + X(2n+4)" in the equation (1). The sum a1 is transferred to the flipflop FF9 of the latch 22. Correspondingly, the adder 21_2, 21_3, and 21_4 calculate the sum a2 of the pixel data m3 and m4, the sum a3 of the pixel data m5 and m6, and the sum a4 of the pixel data m7 and m8, respectively The calculations of sums a2, a2 and a3 are equivalent to the calculations of the term "X(2n+3)+X(2n-3)", "X(2n+2)+X(2n-2)", and "X(2n+1)+X(2n-1)", respectively, in the equation (1). The sums a2, a3, and a4 are transferred to the flipflops FF10, FF11, FF12, respectively.

In the meantime, the flipflop FF13 receives the pixel data f4, associated with the pixel of

interest, from the flipflop FF4.

The controller unit 30 then develops the control signal S3 describing the coefficients α to ε so that the coefficients α to ε are adaptive to encoding through the discrete wavelet transform using the irreversible 9/7 filter. In the event that the pixel of interest is positioned in the even numbered columns, the coefficient α is set to W1, and this results in that the multiplier 23_1 calculates the product MPY1 of the sum a1 and the coefficient W1. The calculation of the product MPY1 is equivalent to the calculation of the term "W1 x {X(2n-4)+X(2n+4)}" in the equation (1). In the event that the pixel of interest is positioned in the even numbered columns, on the other hand, the product MPY1 is set to zero, because the coefficient α is defined as being zero. The product MPY1 is transferred from the multiplier 23_1 to the flipflop FF14 of the latch 24.

Correspondingly, the coefficient β is set to -W0 in the event that the pixel of interest is positioned in the even numbered columns, and this results in that the multiplier 23_2 calculates the product MPY2 of the sum a2 and the coefficient -W0. The calculation of the product MPY2 is equivalent to the calculation of the term "-W0 x

{X(2n-3)+X(2n+3)}" in the equation (1). In the event that the pixel of interest is positioned in the even numbered columns, on the other hand, the coefficient β is set to W4, and this results in that the multiplier 23_2 calculates the product MPY2 of the sum a2 and the coefficient W4. The calculation of the product MPY2 is equivalent to the calculation of the term "W4 x {X(2n-3)+X(2n+3)}" in the equation (1). The product MPY2 is transferred from the multiplier 23_2 to the flipflop FF15 of the latch 24.

Correspondingly, the coefficient γ is set to -W3 in the event that the pixel of interest is positioned in the even numbered columns, and this results in that the multiplier 23_3 calculates the product MPY3 of the sum a3 and the coefficient -W3. The calculation of the product MPY3 is equivalent to the calculation of the term "-W3 x {X(2n-2)+X(2n+2)}" in the equation (1). In the event that the pixel of interest is positioned in the even numbered columns, on the other hand, the coefficient γ is set to -W2, and this results in that the multiplier 23_3 calculates the product MPY3 of the sum a3 and the coefficient -W2. The calculation of the product MPY3 is equivalent to the calculation of the term "-W2 x {X(2n-2)+X(2n+2)}" in the equation (1). The product

MPY3 is transferred from the multiplier 23_3 to the flipflop FF16 of the latch 24.

Correspondingly, the coefficient δ is set to W5 in the event that the pixel of interest is positioned in the even numbered columns, and this results in that the multiplier 23_4 calculates the product MPY4 of the sum a4 and the coefficient W5. The calculation of the product MPY4 is equivalent to the calculation of the term "W5 x {X(2n-1)+X(2n+1)}" in the equation (1). In the event that the pixel of interest is positioned in the even numbered columns, on the other hand, the coefficient δ is set to -W6, and this results in that the multiplier 23_4 calculates the product MPY4 of the sum a4 and the coefficient -W6. The calculation of the product MPY3 is equivalent to the calculation of the term "-W6 x {X(2n-1)+X(2n+1)}" in the equation (1). The product MPY4 is transferred from the multiplier 23_4 to the flipflop FF17 of the latch 24.

Correspondingly, the coefficient ε is set to W7 in the event that the pixel of interest is positioned in the even numbered columns, and this results in that the multiplier 23_5 calculates the product MPY5 of the pixel data f4 from the flipflop FF13 and the coefficient W5. The calculation of the product MPY5 is equivalent to

the calculation of the term "W7 x X(2n)" in the equation (1). In the event that the pixel of interest is positioned in the even numbered columns, on the other hand, the coefficient ε is set to W8, and this results in that the multiplier 23_5 calculates the product MPY5 of the pixel data f4 and the coefficient W8. The calculation of the product MPY5 is equivalent to the calculation of the term "W8 x X(2n+1)" in the equation (1). The product MPY5 is transferred from the multiplier 23_5 to the flipflop FF18 of the latch 24.

The controller unit 30 develops the control signal S4 to indicate the selector 25_5 within the adder/subtractor unit 25 to select the output of the flipflop FF18. This allows the adder/subtractor unit 25 to calculate the sum of the products MPY1 to MPY5 received from the respective flipflops FF14 to FF18 by using the adder 25_1 to adder 25_4. The sum of the products MPY1 to MPY5 is equal to Y(2n) in the equation (1) in the event that the pixel of interest is positioned in the even numbered columns, while equal to Y(2n+1) in the equation (2) in the event that the pixel of interest is positioned in the odd numbered columns. After the calculation of Y(2n) or Y(2n+1), the adder/subtractor unit 25

transfers Y(2n) or Y(2n+1) to the flipflop FF19.

The flipflop FF19 provides the limiter 26 with Y(2n) or Y(2n+1), and the output of the limiter 26 is latched by the flipflop FF20. The output of the flipflop FF20 is the encoded pixel data for the pixel of interest.

The same goes for the other pixels of the image, and this achieves 2-dimentional discrete wavelet transform of the pixel data.

(1-2) Decoding through the discrete wavelet transform using the irreversible 9/7 filter

The procedure of decoding the pixel data through the discrete wavelet transform using the irreversible 9/7 filter is almost identical to that of encoding, except for that the rearrangement of the pixel data f0 to f8 is executed as indicated by the row "9/7 DECODING" in Fig. 3, and that the coefficients α to ε are set to the value as indicated by the second row in Fig. 3. Therefore, no detailed explanation of the decoding is given.

(1-3) Encoding through the discrete cosine transform

Figs. 5 to 7 are timing diagrams illustrating the procedure of encoding the pixel

data through the discrete cosine transform. The procedures at clock periods CLK1 to CLK27, which are defined by a clock signal, are respectively described below in detail.

Clock Periods CLK1 to CLK9

As shown in Fig. 5, the pixel data f0 to f7 are serially transferred to the flipflop FF0 to FF7, respectively, during the clock period CLK1 through CLK9. After the flipflops FF0 to FF7 latches the pixel data f0 to f7, the processing apparatus starts encoding the pixel data of the pixels.

At the clock period CLK9, the controller unit 30 develops the control signal S1 to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK9. In response to the control signal S1, the rearrangement circuit 12 outputs the pixel data f0, and f7 as the pixel data m5, and m6, respectively. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a0 (= f0 + f7) in the equation (9). The calculated sum a0 is stored into the flipflop FF11 at the end of the clock period CLK9. It should be noted that Fig. 5 to 7 refer to invalid data as symbols "*", while referring to zero as symbols "x0".

Clock Period CLK10

At the following clock period CLK10, the rearrangement circuit 12 outputs the pixel data f1 and f6 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK10. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a1 (= f1 + f6) in the equation (9). The calculated sum a1 is stored into the flipflop FF11 at the end of the clock period CLK10.

In the meantime, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a0 (=f0 + f7) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a0 and the coefficient D3 as described in the equation (10). The calculated product "a0 x D3" is stored into the flipflop FF16 at the end of the clock period CLK10.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK10.

Clock Period CLK11

During the following clock period CLK11, the rearrangement circuit 12 outputs the pixel data f2 and f5 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK11. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a2 (= f2 + f5) described in the equation (9). The calculated sum a2 is stored into the flipflop FF11 at the end of the clock period CLK11.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a1 (=f1 + f6) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a1 and the coefficient D3 as described in the equation (10). The calculated product "a1 x D3" is stored into the flipflop FF16 at the end of the clock period CLK11.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK11.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the

flipflops FF14 to FF17 and the selector 25_5. The fact that the flipflops FF14, FF15, and FF17 are reset to zero results in that the output of the adder/subtractor unit 25 is equal to the output of the flipflop FF16, that is, the product "a0 x D3". The product "a0 x D3" is latched by the flipflop FF19 at the end of the CLK11.

Clock Period CLK12

During the following clock period CLK12, the rearrangement circuit 12 outputs the pixel data f3 and f4 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK12. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a3 (= f3 + f4) described in the equation (9). The calculated sum a3 is stored into the flipflop FF11 at the end of the clock period CLK12.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a2 (= f2 + f5) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a2 and the coefficient D3 as described in the equation (10).

The calculated product "a2 x D3" is stored into the flipflop FF16 at the end of the clock period CLK11.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK12.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15, and FF17 are reset to zero results in that the adder/subtractor unit 25 calculates the sum of the product "a0 x D3" received from the flipflop FF19 and the product "a1 x D3" received from the flipflop FF16, that is, the term "(a0 + a1) x D3". The calculated term "(a0 + a1) x D3" is latched by the flipflop FF19 at the end of the CLK12.

Furthermore, the product "a0 x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK12 before the latch of the calculated term "(a0 + a1) x D3".

Clock Period CLK13

During the following clock period CLK13, as shown in Fig. 6, the rearrangement circuit 12

outputs the pixel data f0 and f7 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK13. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a0 (= f0 + f7) described in the equation (9). The calculated sum a3 is stored into the flipflop FF11 at the end of the clock period CLK13.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a3 (=f3 + f4) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a3 and the coefficient D3 as described in the equation (10). The calculated product "a3 x D3" is set to the flipflop FF16 at the end of the clock period CLK13.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK13.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15,

and FF17 are reset to zero results in that the adder/subtractor unit 25 calculates the sum of the product "(a0 + a1) x D3" received from the flipflop FF19 and the product "a2 x D3" received from the flipflop FF16, that is, the term "(a0 + a1 + a2) x D3". The calculated term "(a0 + a1 + a2) x D3" is latched by the flipflop FF19 at the end of the CLK13.

Furthermore, the term "(a0 + a1) x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK13 before the latch of the calculated term "(a0 + a1 + a2) x D3".

Clock Period CLK14

During the following clock period CLK14, the rearrangement circuit 12 outputs the pixel data f1 and f6 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK14. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum -a1 (= -(f1 + f6)) described in the equation (9). The calculated sum -a1 is stored into the flipflop FF11 at the end of the clock period CLK14.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a0 (=f0 + f7) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a0 and the coefficient D3 as described in the equation (10). The calculated product "a0 x D3" is stored into the flipflop FF16 at the end of the clock period CLK14.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK14.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15, and FF17 are reset to zero results in that the adder/subtractor unit 25 calculates the sum of the product "(a0 + a1 + a2) x D3" received from the flipflop FF19 and the product "a3 x D3" received from the flipflop FF16, that is, the term "(a0 + a1 + a2 + a3) x D3". The calculated term "(a0 + a1 + a2 + a3) x D3" is latched by the flipflop FF19 at the end of the CLK14.

Furthermore, the term "(a0 + a1 +a2) x D3", which has been latched by the flipflop FF19, is

transferred to the flipflop FF20 at the end of the clock period CLK14 before the latch of the calculated term "(a0 + a1 + a2 +a3) x D3".

Clock Period CLK15

During the following clock period CLK15, the rearrangement circuit 12 outputs the pixel data f2 and f5 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK15. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum -a2 (= -(f2 + f5)) described in the equation (9). The calculated sum -a2 is stored into the flipflop FF11 at the end of the clock period CLK15.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum -a1 (= -(f1 + f6)) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum -a1 and the coefficient D3 as described in the equation (10). The calculated product "-a1 x D3" is stored into the flipflop FF16 at the end of the clock period CLK15.

In addition, the flipflops FF14, FF15, and

FF17 are reset to zero at the end of the clock period CLK15.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 is controlled to output zero, and the flipflops FF14, FF15, and FF17 are reset to zero results in that the adder/subtractor unit 25 outputs the product "a0 x D3", which has been stored in the flipflop FF16. The calculated product "a0 x D3" is latched by the flipflop FF19 at the end of the CLK15.

Furthermore, the term "(a0 + a1 + a2 +a3) x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK15 before the latch of the calculated produce "a0 x D3". This allows the output of the encoded pixel data F0 (=(a0 + a1 + a2 +a3) x D3) from the flipflop FF20 at the following clock period CLK16.

Clock Period CLK16

During the following clock period CLK16, the rearrangement circuit 12 outputs the pixel data f3 and f4 as the pixel data m5 and m6, respectively, in response to the control signal S1, which is developed to indicate the

rearrangement circuit 12 to execute a procedure defined for the clock period CLK16. The adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a3 (= f3 + f4) described in the equation (9). The calculated sum a3 is stored into the flipflop FF11 at the end of the clock period CLK16.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum -a2 (= -(f2 + f5)) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum -a2 and the coefficient D3 as described in the equation (10). The calculated product "-a2 x D3" is stored into the flipflop FF16 at the end of the clock period CLK16.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK16.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15, and FF17 are reset to zero results in that the adder/subtractor unit 25 calculates the sum of the product "a0 x D3" received from the flipflop

F19 and the product "-a1 x D3" received from the flipflop FF16, that is, the term "(a0 - a1) x D3". The calculated term "(a0 - a1) x D3" is latched by the flipflop FF19 at the end of the CLK16.

Furthermore, the term "a0 x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK16 before the latch of the calculated term "(a0 - a1) x D3".

Clock Period CLK17

During the following clock period CLK17, the rearrangement circuit 12 outputs the pixel data f0 and f7 as the pixel data m1 and m2, the pixel data f1 and f6 as the pixel data m3 and m4, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK17. The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum a0 (= f0 + f7) described in the equation (9). The calculated sum a0 is stored in the flipflop FF9 at the end of the clock period CLK17. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a1 (= f1 + f6) described in the equation (9). The calculated sum a1 is stored into the flipflop

FF10 at the end of the clock period CLK17.

In the mean time, the controller unit 30 sets the coefficient γ to D3, and the multiplier 23_3 receives the sum a3 (= f3 + f4) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a2 and the coefficient D3 as described in the equation (10). The calculated product "a3 x D3" is stored into the flipflop FF16 at the end of the clock period CLK17.

In addition, the flipflops FF14, FF15, and FF17 are reset to zero at the end of the clock period CLK17.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15, and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the term "(a0 -a1) x D3" received from the flipflop F19 and the product "-a2 x D3" received from the flipflop FF16, that is, the term "(a0 - a1 - a2) x D3". The calculated term "(a0 - a1 - a2) x D3" is latched by the flipflop FF19 at the end of the CLK17.

Furthermore, the term "(a0 - a1) x D3",

which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK17 before the latch of the calculated term "(a0 - a1 - a2) x D3".

Clock Period CLK18

During the following clock period CLK18, the rearrangement circuit 12 outputs the pixel data f3 and f4 as the pixel data m1 and m2, the pixel data f2 and f5 as the pixel data m3 and m4, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK18. The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum -a3 (= -(f3 + f4)) described in the equation (9). The calculated sum -a3 is stored in the flipflop FF9 at the end of the clock period CLK18. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum -a2 (= -(f2 + f5)) described in the equation (9). The calculated sum -a3 is stored into the flipflop FF10 at the end of the clock period CLK18.

In the mean time, the controller unit 30 sets the coefficient α to D5, and the multiplier 23_1 receives the sum a0 (= f0 + f7) from the

flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum a0 and the coefficient D5 as described in the equation (10). The calculated product "a0 x D5" is stored into the flipflop FF14 at the end of the clock period CLK18.

Correspondingly, the controller unit 30 sets the coefficient β to D1, and the multiplier 23_2 receives the sum a1 (= f1 + f6) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a1 and the coefficient D1 as described in the equation (10). The calculated product "a1 x D1" is stored into the flipflop FF15 at the end of the clock period CLK18.

In addition, the flipflops FF16, and FF17 are reset to zero at the end of the clock period CLK18.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF14, FF15, and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the term "(a0 - a1 - a2) x D3" received from the flipflop F19 and the product "a3 x D3" received

from the flipflop FF16, that is, the term "(a0 - a1 - a2 + a3) x D3". The calculated term "(a0 - a1 -a2 + a3) x D3" is latched by the flipflop FF19 at the end of the CLK18.

Furthermore, the term "(a0 - a1 - a2) x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK18 before the latch of the calculated term "(a0 - a1 - a2 + a3) x D3".

Clock Period CLK19

During the following clock period CLK19, the rearrangement circuit 12 outputs the pixel data f1 and f6 as the pixel data m1 and m2, the pixel data f0 and f7 as the pixel data m3 and m4, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK19. The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum -a1 (= -(f1 + f6)) described in the equation (9). The calculated sum -a1 is stored into the flipflop FF9 at the end of the clock period CLK19. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a0 (= f0 + f7) described in the equation (9). The calculated sum a0 is stored

into the flipflop FF10 at the end of the clock period CLK19.

In the mean time, the controller unit 30 sets the coefficient α to D5, and the multiplier 23_1 receives the sum -a3 (= -(f3 + f4)) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum -a3 and the coefficient D5 as described in the equation (10). The calculated product "-a3 x D5" is stored into the flipflop FF14 at the end of the clock period CLK19.

Correspondingly, the controller unit 30 sets the coefficient β to D1, and the multiplier 23_2 receives the sum -a2 (= -(f2 + f5) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum -a2 and the coefficient D1 as described in the equation (10). The calculated product "-a2 x D1" is stored into the flipflop FF15 at the end of the clock period CLK19.

In addition, the flipflops FF16, and FF17 are reset to zero at the end of the clock period CLK19.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the

output by the controller unit 30. The fact that the selector 25_5 outputs zero, and the flipflops FF16, and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the product "a0 x D5" received from the flipflop FF14 and the product "a1 x D1" received from the flipflop FF15, that is, the term "a0 x D4 + a1 x D1". The calculated term "a0 x D4 + a1 x D1" is latched by the flipflop FF19 at the end of the CLK19.

Furthermore, the term "(a0 - a1 - a2 + a3) x D3", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK19 before the latch of the calculated term "a0 x D4 + a1 x D1". This allows the output of the encoded pixel data F4 (=(a0 - a1 - a2 +a3) x D3) from the flipflop FF20 at the following clock period CLK20.

Clock Period CLK20

During the following clock period CLK20, the rearrangement circuit 12 outputs the pixel data f2 and f5 as the pixel data m1 and m2, and the pixel data f3 and f4 as the pixel data m3 and m4, respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure

defined for the clock period CLK20. The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum a2 (= f2 + f5) described in the equation (9). The calculated sum a2 is stored into the flipflop FF9 at the end of the clock period CLK20. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum -a3 (= -(f3 + f4) described in the equation (9). The calculated sum -a3 is stored into the flipflop FF10 at the end of the clock period CLK20.

In the mean time, the controller unit 30 sets the coefficient α to D5, and the multiplier 23_1 receives the sum -a1 (= -(f1 + f6)) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum -a1 and the coefficient D5 as described in the equation (10). The calculated product "-a1 x D5" is stored into the flipflop FF14 at the end of the clock period CLK20.

Correspondingly, the controller unit 30 sets the coefficient β to D1, and the multiplier 23_2 receives the sum a0 (= f0 + f7) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a0 and the coefficient D1 as described in the equation (10). The calculated product "a0 x D1" is stored into

the flipflop FF15 at the end of the clock period CLK20.

In addition, the flipflops FF16, and FF17 are reset to zero at the end of the clock period CLK20.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF16 and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the term "a0 x D5 + a1 x D1" received from the flipflop FF19, the product "-a3 x D5" received from the flipflop FF14 and the product "-a2 x D1" received from the flipflop FF15, that is, the term "(a0 - a3) x D5 + (a1 - a2) x D1". The calculated term "(a0 - a3) x D5 + (a1 - a2) x D1" is latched by the flipflop FF19 at the end of the CLK20.

Furthermore, the term "a0 x D5 + a1 x D1", which has been latched by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK20 before the latch of the calculated term "(a0 - a3) x D5 + (a1 - a2) x D1".

Clock Period CLK21

During the following clock period CLK21, as shown in Fig. 7, the rearrangement circuit 12 outputs the pixel data f0 and f7 as the pixel data m1 and m2, the pixel data f1 and f6 as the pixel data m3 and m4, the pixel data f2 and f5 as the pixel data m5 and m6, and the pixel data f3 and f4 as the pixel data m7 and m8 respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK21.

The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum a4 (= f0 - f7) described in the equation (9). The calculated sum a4 is stored into the flipflop FF9 at the end of the clock period CLK21. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a5 (= f1 - f6) described in the equation (9). The calculated sum a5 is stored into the flipflop FF10 at the end of the clock period CLK21. Furthermore, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a6 (= f2 - f5) described in the equation (9). The calculated sum a6 is stored into the flipflop FF11 at the end of the clock period CLK21. In addition, the adder 21_4 calculates the sum of the pixel data m7 and m8,

that is, the sum a7 (= f3 - f4) described in the equation (9). The calculated sum a7 is stored into the flipflop FF12 at the end of the clock period CLK21.

In the mean time, the controller unit 30 sets the coefficient α to D5, and the multiplier 23_1 receives the sum a2 (= f2 + f5) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum a2 and the coefficient D5 as described in the equation (10). The calculated product "a2 x D5" is stored into the flipflop FF14 at the end of the clock period CLK21.

Correspondingly, the controller unit 30 sets the coefficient β to D1, and the multiplier 23_2 receives the sum -a3 (= f0 + f7) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum -a3 and the coefficient D1 as described in the equation (10). The calculated product "-a3 x D1" is stored into the flipflop FF15 at the end of the clock period CLK21.

In addition, the flipflops FF16, and FF17 are reset to zero at the end of the clock period CLK21.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the

flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. The fact that the selector 25_5 outputs zero and the flipflops FF16 and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the product "-a1 x D5" received from the flipflop FF14 and the product "a0 x D1" received from the flipflop FF15, that is, the term "-a1 x D5 + a0 x D1". The calculated term "-a1 x D5 + a0 x D1" is latched by the flipflop FF19 at the end of the CLK21.

Furthermore, the term "(a0 - a3) x D5 + (a1 - a2) x D1", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK21 before the latch of the calculated term "-a1 x D5 + a0 x D1". This allows the output of the encoded pixel data F2 (=(a0 - a3) x D5 + (a1 - a2) x D1) from the flipflop FF20 at the following clock period CLK22.

Clock Period CLK22

During the following clock period CLK22, the rearrangement circuit 12 outputs the pixel data f2 and f5 as the pixel data m1 and m2, the pixel data f0 and f7 as the pixel data m3 and m4,

the pixel data f3 and f4 as the pixel data m5 and m6, and the pixel data f1 and f6 as the pixel data m7 and m8 respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK22.

The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum -a6 (= -(f2 - f5)) described in the equation (9). The calculated sum -a6 is stored into the flipflop FF9 at the end of the clock period CLK22. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a4 (= f0 - f7) described in the equation (9). The calculated sum a4 is stored into the flipflop FF10 at the end of the clock period CLK22. Furthermore, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum -a7 (= -(f3 - f4)) described in the equation (9). The calculated sum -a7 is stored into the flipflop FF11 at the end of the clock period CLK22. In addition, the adder 21_4 calculates the sum of the pixel data m7 and m8, that is, the sum -a5 (= -(f1 - f6) described in the equation (9). The calculated sum -a5 is stored into the flipflop FF12 at the end of the clock period CLK22.

In the mean time, the controller unit 30

sets the coefficient α to D6, and the multiplier 23_1 receives the sum a4 (= f0 - f7) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum a4 and the coefficient D6 as described in the equation (10). The calculated product "a4 x D6" is stored into the flipflop FF14 at the end of the clock period CLK22.

Correspondingly, the controller unit 30 sets the coefficient β to D4, and the multiplier 23_2 receives the sum a4 (= f0 - f7) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a4 and the coefficient D4 as described in the equation (10). The calculated product "a4 x D4" is stored into the flipflop FF15 at the end of the clock period CLK22.

Correspondingly, the controller unit 30 sets the coefficient γ to D2, and the multiplier 23_3 receives the sum a6 (= f2 - f5) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a6 and the coefficient D2 as described in the equation (10). The calculated product "a6 x D2" is stored into the flipflop FF16 at the end of the clock period CLK22.

Correspondingly, the controller unit 30 set

the coefficient δ to D0, and the multiplier 23_4 receives the sum a7 (= f3 - f4) from the flipflop FF12. This allows the multiplier 23_4 to calculate the product of the sum a7 and the coefficient D0 as described in the equation (10). The calculated product "a7 x D0" is stored into the flipflop FF17 at the end of the clock period CLK22.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to select the output of the flipflop FF19 by the controller unit 30. The fact that the selector 25_5 selects the output of the flipflop FF19, and the flipflops FF16 and FF17 are reset to zero, results in that the adder/subtractor unit 25 calculates the sum of the term "-a1 x D5 + a0 x D1" received from the flipflop FF19, the product "a2 x D5" received from the flipflop FF14, and the product "-a3 x D1" received from the flipflop FF15, that is, the term "(-a1 + a2) x D5 + (a0 - a3) x D1". The calculated term "(-a1 + a2) x D5 + (a0 - a3) x D1" is latched by the flipflop FF19 at the end of the CLK22.

Furthermore, the term "-a1 x D5 + a0 x D1", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of

the clock period CLK22 before the latch of the calculated term "(-a1 + a2) x D5 + (a0 - a3) x D1".

Clock Period CLK23

During the following clock period CLK23, the rearrangement circuit 12 outputs the pixel data f1 and f6 as the pixel data m1 and m2, the pixel data f3 and f4 as the pixel data m3 and m4, the pixel data f0 and f7 as the pixel data m5 and m6, and the pixel data f2 and f5 as the pixel data m7 and m8 respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK23.

The adder 21_1 calculates the sum of the pixel data m1 and m2, that is the sum -a5 (= -(f1 - f6)) described in the equation (9). The calculated sum -a5 is stored into the flipflop FF9 at the end of the clock period CLK23. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a7 (= f3 - f4) described in the equation (9). The calculated sum a7 is stored into the flipflop FF10 at the end of the clock period CLK23. Furthermore, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum a4 (= f0 - f7) described

in the equation (9). The calculated sum a4 is stored into the flipflop FF11 at the end of the clock period CLK23. In addition, the adder 21_4 calculates the sum of the pixel data m7 and m8, that is, the sum a6 (= f2 - f5) described in the equation (9). The calculated sum a6 is stored into the flipflop FF12 at the end of the clock period CLK23.

In the mean time, the controller unit 30 sets the coefficient α to D6, and the multiplier 23_1 receives the sum -a6 (= -(f2 - f5)) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum -a6 and the coefficient D6 as described in the equation (10). The calculated product "-a6 x D6" is stored into the flipflop FF14 at the end of the clock period CLK23.

Correspondingly, the controller unit 30 sets the coefficient β to D4, and the multiplier 23_2 receives the sum a0 (= f0 - f7) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a0 and the coefficient D4 as described in the equation (10). The calculated product "a0 x D4" is stored into the flipflop FF15 at the end of the clock period CLK23.

Correspondingly, the controller unit 30

sets the coefficient γ to D2, and the multiplier 23_3 receives the sum -a7 (= -(f3 - f4)) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum -a7 and the coefficient D2 as described in the equation (10). The calculated product "-a7 x D2" is stored into the flipflop FF16 at the end of the clock period CLK23.

Correspondingly, the controller unit 30 set the coefficient δ to D0, and the multiplier 23_4 receives the sum -a5 (= -(f1 - f6)) from the flipflop FF12. This allows the multiplier 23_4 to calculate the product of the sum -a5 and the coefficient D0 as described in the equation (10). The calculated product "-a5 x D0" is stored into the flipflop FF17 at the end of the clock period CLK23.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the sum of the product "a4 x D6" received from the flipflop FF14, the product "a5 x D4" received from the flipflop FF15, the product "a6 x D2" received from the flipflop FF16, and the product

"a7 x D0" received from the flipflop FF17, that is, the term "a4 x D6 + a5 x D4 + a6 x D2 + a7 x D0". The calculated term "a4 x D6 + a5 x D4 + a6 x D2 + a7 x D0" is latched by the flipflop FF19 at the end of the CLK23.

Furthermore, the term "(-a1 + a2) x D5 + (a0 - a3) x D1", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK23 before the latch of the calculated term "a4 x D6 + a5 x D4 + a6 x D2 + a7 x D0". This allows the output of the encoded pixel data F6 (=(-a1 + a2) x D5 + (a0 - a3) x D1) from the flipflop FF20 at the following clock period CLK24.

Clock Period CLK24

During the following clock period CLK24, the rearrangement circuit 12 outputs the pixel data f3 and f4 as the pixel data m1 and m2, the pixel data f2 and f5 as the pixel data m3 and m4, the pixel data f1 and f6 as the pixel data m5 and m6, and the pixel data f0 and f7 as the pixel data m7 and m8 respectively, in response to the control signal S1, which is developed to indicate the rearrangement circuit 12 to execute a procedure defined for the clock period CLK24.

The adder 21_1 calculates the sum of the

pixel data m1 and m2, that is the sum -a7 (= -(f3 - f4)) described in the equation (9). The calculated sum -a7 is stored into the flipflop FF9 at the end of the clock period CLK24. In the meantime, the adder 21_2 calculates the sum of the pixel data m3 and m4, that is, the sum a6 (= f2 - f5) described in the equation (9). The calculated sum a6 is stored into the flipflop FF10 at the end of the clock period CLK24. Furthermore, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum -a5 (= f1 - f6) described in the equation (9). The calculated sum -a5 is stored into the flipflop FF11 at the end of the clock period CLK24. In addition, the adder 21_4 calculates the sum of the pixel data m7 and m8, that is, the sum a4 (= f0 - f7) described in the equation (9). The calculated sum a4 is stored into the flipflop FF12 at the end of the clock period CLK24.

In the mean time, the controller unit 30 sets the coefficient α to D6, and the multiplier 23_1 receives the sum -a5 (= -(f1 - f6)) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum -a5 and the coefficient D6 as described in the equation (10). The calculated product "-a5 x D6" is stored into the flipflop FF14 at the end of the clock period

CLK24.

Correspondingly, the controller unit 30 sets the coefficient β to D4, and the multiplier 23_2 receives the sum a7 (= f3 - f4) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a7 and the coefficient D4 as described in the equation (10). The calculated product "a7 x D4" is stored into the flipflop FF15 at the end of the clock period CLK24.

Correspondingly, the controller unit 30 sets the coefficient γ to D2, and the multiplier 23_3 receives the sum a4 (= f0 - f7) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum a4 and the coefficient D2 as described in the equation (10). The calculated product "a4 x D2" is stored into the flipflop FF16 at the end of the clock period CLK24.

Correspondingly, the controller unit 30 set the coefficient δ to D0, and the multiplier 23_4 receives the sum a6 (= f2 - f5) from the flipflop FF12. This allows the multiplier 23_4 to calculate the product of the sum a6 and the coefficient D0 as described in the equation (10). The calculated product "a6 x D0" is stored into the flipflop FF17 at the end of the clock period CLK24.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the sum of the product "-a6 x D6" received from the flipflop FF14, the product "-a4 x D4" received from the flipflop FF15, the product "-a7 x D2" received from the flipflop FF16, and the product "-a5 x D0" received from the flipflop FF17, that is, the term "-a6 x D6 + a4 x D4 - a7 x D2 - a5 x D0". The calculated term "-a6 x D6 + a4 x D4 - a7 x D2 - a5 x D0" is latched by the flipflop FF19 at the end of the CLK24.

Furthermore, the term "a4 x D6 + a5 x D4 + a6 x D2 + a7 x D0", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK24 before the latch of the term "-a6 x D6 + a4 x D4 - a7 x D2 - a5 x D0". This allows the output of the encoded pixel data F1 (= a4 x D6 + a5 x D4 + a6 x D2 + a7 x D0) from the flipflop FF20 at the following clock period CLK25.

Clock Period CLK25

At the following clock period CLK25, pixel

data of a next pixel of interest are provided for the flipflops FF0 to FF7. The procedure for encoding the pixel data of the next pixel of interest is identical to that of the pixel data of the current pixel of interest. Therefore, detailed explanation is not given, hereinafter, for encoding the pixel data of the next pixel of interest.

During the clock period CLK25, the rearrangement circuit 12 outputs the pixel data m1 to m8 for the next pixel of interest in response to the control signal S1 received from the controller unit 30.

In the mean time, the controller unit 30 sets the coefficient α to D6, and the multiplier 23_1 receives the sum -a7 (= -(f3 - f4)) from the flipflop FF9. This allows the multiplier 23_1 to calculate the product of the sum -a7 and the coefficient D6 as described in the equation (10). The calculated product "-a7 x D6" is stored into the flipflop FF14 at the end of the clock period CLK25.

Correspondingly, the controller unit 30 sets the coefficient β to D4, and the multiplier 23_2 receives the sum a6 (= f2 - f5) from the flipflop FF10. This allows the multiplier 23_2 to calculate the product of the sum a6 and the

coefficient D4 as described in the equation (10). The calculated product "a6 x D4" is stored into the flipflop FF15 at the end of the clock period CLK25.

Correspondingly, the controller unit 30 sets the coefficient γ to D2, and the multiplier 23_3 receives the sum -a5 (= f1 - f6) from the flipflop FF11. This allows the multiplier 23_3 to calculate the product of the sum -a5 and the coefficient D2 as described in the equation (10). The calculated product "-a5 x D2" is stored into the flipflop FF16 at the end of the clock period CLK25.

Correspondingly, the controller unit 30 set the coefficient δ to D0, and the multiplier 23_4 receives the sum a4 (= f0 - f7) from the flipflop FF12. This allows the multiplier 23_4 to calculate the product of the sum a4 and the coefficient D0 as described in the equation (10). The calculated product "a4 x D0" is stored into the flipflop FF17 at the end of the clock period CLK25.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the

sum of the product "-a5 x D6" received from the flipflop FF14, the product "-a7 x D4" received from the flipflop FF15, the product "a4 x D2" received from the flipflop FF16, and the product "a6 x D0" received from the flipflop FF17, that is, the term "-a5 x D6 - a7 x D4 + a4 x D2 + a6 x D0". The calculated term "-a5 x D6 - a7 x D4 + a4 x D2 + a6 x D0" is latched by the flipflop FF19 at the end of the CLK25.

Furthermore, the term "-a6 x D6 + a4 x D4 - a7 x D2 - a5 x D0", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK25 before the latch of the term "-a5 x D6 - a7 x D4 + a4 x D2 + a6 x D0". This allows the output of the encoded pixel data F3 (= -a6 x D6 + a4 x D4 - a7 x D2 - a5 x D0) from the flipflop FF20 at the following clock period CLK26.

Clock Period CLK26

During the clock period CLK25, the rearrangement circuit 12 outputs the pixel data m1 to m8 for the next pixel of interest in response to the control signal S1 received from the controller unit 30. In the mean time, the multipliers 23_1 to 23_4 execute the operation for encoding the pixel data of the next pixel of

interest.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the sum of the product "-a7 x D6" received from the flipflop FF14, the product "-a6 x D4" received from the flipflop FF15, the product "-a5 x D2" received from the flipflop FF16, and the product "a4 x D0" received from the flipflop FF17, that is, the term "-a7 x D6 - a6 x D4 - a5 x D2 + a4 x D0". The calculated term "-a7 x D6 - a6 x D4 - a5 x D2 + a4 x D0" is latched by the flipflop FF19 at the end of the CLK26.

Furthermore, the term "-a5 x D6 + a7 x D4 + a4 x D2 + a6 x D0", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK26 before the latch of the term "-a7 x D6 - a6 x D4 - a5 x D2 + a4 x D0". This allows the output of the encoded pixel data F5 (= -a5 x D6 + a7 x D4 + a4 x D2 + a6 x D0) from the flipflop FF20 at the following clock period CLK27.

Clock Period CLK27

At the following clock period CLK27, the rearrangement circuit 12 outputs the pixel data m1 to m8 for the next pixel of interest in response to the control signal S1 received from the controller unit 30. In the mean time, the multipliers 23_1 to 23_4 execute the operation for encoding the pixel data of the next pixel of interest. Furthermore, the adder/subtractor unit 25 executes the operation for encoding the pixel data of the next pixel of interest.

During the clock period CLK27, the term "-a7 x D6 + a6 x D4 - a5 x D2 + a4 x D0", which has been stored by the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK27. This allows the output of the encoded pixel data F7 (= -a7 x D6 + a6 x D4 - a5 x D2 + a4 x D0) from the flipflop FF20 at the following clock period CLK28.

(1-4) Decoding through the discrete cosine transform

Figs. 8 to 10 are timing diagram illustrating the procedure of decoding the pixel data through the discrete cosine transform.

Referring to the equations (13) and (14), the decoded pixel data f0 is obtained from the following equation:

$$f0 = \{(f0 + f4) /2\} + \{(f0 - f4) /2\}. \quad (15)$$

The right hand first term of the equation (15) is obtained from the first row of the matrix of the equation (13), while the right hand second term of the equation (15) is obtained from the first row of the matrix of the equation (14).

Correspondingly, the decoded pixel data f1 to f7 are obtained from the following equations:

$$f1 = \{(f1 + f5) /2\} + \{(f1 - f5) /2\}, \quad (17)$$

$$f2 = \{(f2 + f6) /2\} + \{(f2 - f6) /2\}, \quad (18)$$

$$f3 = \{(f3 + f7) /2\} + \{(f3 - f7) /2\}, \quad (19)$$

$$f4 = \{(f0 + f4) /2\} - \{(f0 - f4) /2\}, \quad (20)$$

$$f5 = \{(f1 + f5) /2\} - \{(f1 - f5) /2\}, \quad (21)$$

$$f6 = \{(f2 + f6) /2\} - \{(f2 - f6) /2\}, \text{ and} \quad (22)$$

$$f7 = \{(f3 + f7) /2\} + \{(f3 - f7) /2\}. \quad (23)$$

Because the procedure of obtaining the decoded pixel data f0 to f7 are almost same, the explanation directed to only the decoded pixel data f0 and f4 will be given.

Clock Periods CLK1 to CLK9

The procedure begins with the input of the encoded pixel data F0 to F7. As shown in Fig. 8, the encoded pixel data F0 to F7 are serially transferred to the flipflops FF0 to FF7, respectively.

At the clock period CLK9, the controller

unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK9. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F2, F6, F0, and F4, as the pixel data m1, m3, m5 and m6, respectively. The pixel data m2, m4, m7, and m8 are set to zero.

The adder 21_1 transfers the pixel data m1 to the flipflop FF9, because the pixel data m2 is set to zero. This results in that the pixel data F2 is stored in the flipflop FF9 at the end of the clock period CLK9. The pixel data F2 is used as the element "a2" in the equation (13).

Correspondingly, the adder 21_2 transfers the pixel data m3 to the flipflop FF10. This results in that the pixel data F6 is stored in the flipflop FF10 at the end of the clock period CLK9. The pixel data F6 is used as the element "a1" in the equation (13).

Furthermore, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum of the pixel data F0 and F4. This results in that the sum "F0 + F4" is stored in the flipflop FF11 at the end of the clock period CLK9. The sum "F0 + F4" is used as the element "a0 + a3" in the equation (13).

Clock Period CLK10

At the following clock period CLK10, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK10. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F1, F3, F5, and F7 as the pixel data m1, m3, m5 and m7, respectively.

The adder 21_1 transfers the pixel data m1 to the flipflop FF9. This results in that the pixel data F1 is stored in the flipflop FF9 at the end of the clock period CLK10. The pixel data F1 is used as the element "a5" in the equation (14).

Correspondingly, the adder 21_2 transfers the pixel data m3 to the flipflop FF10. This results in that the pixel data F3 is stored in the flipflop FF10 at the end of the clock period CLK10. The pixel data F3 is used as the element "a7" in the equation (14).

On the other hand, the adder 21_3 inverts the sign of the pixel data m5. The sign-inverted pixel data -m5 is transferred to the flipflop FF11. This results in that the sign-inverted pixel data -F5 is stored in the flipflop FF11 at

the end of the clock period CLK10. The sign-inverted pixel data -F5 is used as the element "-a6" in the equation (14).

Correspondingly, the adder 21_4 inverts the sign of the pixel data m7. The sign-inverted pixel data -m7 is transferred to the flipflop FF11. This results in that the sign-inverted pixel data -F5 is stored in the flipflop FF12 at the end of the clock period CLK10. The sign-inverted pixel data -F5 is used as the element "-a4" in the equation (14).

In the meantime, the multiplier 23_1 receives the pixel data F2(=a2) from the flipflop FF9, while the controller unit 30 sets the coefficient α to D5. This allows the multiplier 23_1 to calculate the product of a2 and D5 used in the equation (13). The product "a2 x D5" is stored into the flipflop FF14 at the end of the clock period CLK10.

Correspondingly, the multiplier 23_2 receives the pixel data F6(=a1) from the flipflop FF10, while the controller unit 30 sets the coefficient β to D1. This allows the multiplier 23_2 to calculate the product of a1 and D1 used in the equation (13). The product "a1 x D1" is stored into the flipflop FF15 at the end of the clock period CLK10.

Correspondingly, the multiplier 23_3 receives the sum "F0 + F4" (= a0 + a3) from the flipflop FF11, while the controller unit 30 sets the coefficient γ to D3. This allows the multiplier 23_3 to calculate the product of the sum "a0 + a3" and the coefficient D1 used in the equation (13). The product "(a0 + a3) x D1" is stored into the flipflop FF16 at the end of the clock period CLK10.

In addition, the flipflop FF17 is reset to zero at the end of the clock period CLK10.

Clock Period CLK11

At the following clock period CLK11, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK11. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F2, F6, F0, and F4 as the pixel data m1, m3, m5 and m6. respectively.

The adder 21_1 transfers the pixel data m1 to the flipflop FF9. This results in that the pixel data F2 is stored in the flipflop FF9 at the end of the clock period CLK11. The pixel data F2 is used as the element "a2" in the equation (13).

Correspondingly, the adder 21_2 transfers the pixel data m3 to the flipflop FF10. This results in that the pixel data F6 is stored in the flipflop FF10 at the end of the clock period CLK11. The pixel data F6 is used as the element "a1" in the equation (14).

On the other hand, the adder 21_3 calculates the sum of the pixel data m5 and m6, that is, the sum of the pixel data F0 and F4. This results in that the sum "F0 + F4" is stored in the flipflop FF11 at the end of the clock period CLK11. The sum "F0 + F4" is used as the element "a0 + a3" in the equation (13).

In the meantime, the multiplier 23_1 receives the pixel data F1(=a5) from the flipflop FF9, while the controller unit 30 sets the coefficient α to D6. This allows the multiplier 23_1 to calculate the product of a5 and D6 used in the equation (14). The product "a5 x D6" is stored into the flipflop FF14 at the end of the clock period CLK11.

Correspondingly, the multiplier 23_2 receives the pixel data F3(=a7) from the flipflop FF10, while the controller unit 30 sets the coefficient β to D4. This allows the multiplier 23_2 to calculate the product of a7 and D4 used in the equation (14). The product "a7 x D4" is

stored into the flipflop FF15 at the end of the clock period CLK11.

Correspondingly, the multiplier 23_3 receives the sign-inverted pixel data -F5(= -a6) from the flipflop FF11, while the controller unit 30 sets the coefficient γ to D2. This allows the multiplier 23_3 to calculate the product of -a6 and D2 used in the equation (14). The product "-a0 x D2" is stored into the flipflop FF16 at the end of the clock period CLK11.

Correspondingly, the multiplier 23_4 receives the sign-inverted pixel data -F7(= -a4) from the flipflop FF12, while the controller unit 30 sets the coefficient δ to D0. This allows the multiplier 23_4 to calculate the product of -a4 and D0 used in the equation (14). The product "-a4 x D0" is stored into the flipflop FF17 at the end of the clock period CLK11.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5, and transfers the calculated sum to the flipflop FF19. The selector 25_5 is controlled to develop zero on the output by the controller unit 30. The fact that the selector 25_5 and the flipflop FF17 output zero results in that the adder/subtractor unit 25 calculates the term "(a0 + a3) x D3 + a2 x D5 +

a1 x D1", that is, the term "(f0 + f4) /2". The calculated term "(f0 + f4) /2" is latched by the flipflop FF19 at the end of the CLK11.

Clock Period CLK12

At the following clock period CLK12, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK12. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F1, F3, F5, and F7 as the pixel data m1, m3, m5 and m7. respectively.

The adder 21_1 transfers the pixel data m1 to the flipflop FF9. This results in that the pixel data F1 is stored in the flipflop FF9 at the end of the clock period CLK12. The pixel data F1 is used as the element "a5" in the equation (14).

Correspondingly, the adder 21_2 transfers the pixel data m3 to the flipflop FF10. This results in that the pixel data F3 is stored in the flipflop FF10 at the end of the clock period CLK12. The pixel data F3 is used as the element "a7" in the equation (14).

On the other hand, the adder 21_3 inverts the sign of the pixel data m5. The sign-inverted

pixel data -m5 is transferred to the flipflop FF11. This results in that the sign-inverted pixel data -F5 is stored in the flipflop FF11 at the end of the clock period CLK12. The sign-inverted pixel data -F5 is used as the element "-a6" in the equation (14).

Correspondingly, the adder 21_4 inverts the sign of the pixel data m7. The sign-inverted pixel data -m7 is transferred to the flipflop FF11. This results in that the sign-inverted pixel data -F5 is stored in the flipflop FF12 at the end of the clock period CLK12. The sign-inverted pixel data -F5 is used as the element "-a4" in the equation (14).

In the meantime, the multiplier 23_1 receives the pixel data F2(=a2) from the flipflop FF9, while the controller unit 30 sets the coefficient α to D5. This allows the multiplier 23_1 to calculate the product of a2 and D5 used in the equation (13). The product "a2 x D5" is stored into the flipflop FF14 at the end of the clock period CLK12.

Correspondingly, the multiplier 23_2 receives the pixel data F6(=a1) from the flipflop FF10, while the controller unit 30 sets the coefficient β to D1. This allows the multiplier 23_2 to calculate the product of a1 and D1 used in

the equation (13). The product "a1 x D1" is stored into the flipflop FF15 at the end of the clock period CLK12.

Correspondingly, the multiplier 23_3 receives the sum "F0 + F4" (= a0 + a3) from the flipflop FF11, while the controller unit 30 sets the coefficient γ to D3. This allows the multiplier 23_3 to calculate the product of the sum "a0 + a3" and the coefficient D1 used in the equation (13). The product "(a0 + a3) x D1" is stored into the flipflop FF16 at the end of the clock period CLK12.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to select the output of the flipflop FF19 by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the decoded pixel data f0, because the adder/subtractor unit 25 calculates the sum of the value "a5 x D6 + a7 x D4 - a6 x D2 - a4 x D0", which is the sum of the outputs of the flipflops FF14 to FF17, and the value "(a0 + a3) x D3 + a2 x D5 + a1 x D1", which is the output of the flipflop FF19. This calculation is equivalent to the calculation of the sum of the term "(f0 + f4) /2" and the term "(f0 - f4) /2". The decoded

pixel data f0 is latched by the flipflop FF19 at the end of the clock period CLK12.

Furthermore, the value "(a0 + a3) x D3 + a2 x D5 + a1 x D1", which has been stored in the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK12. It should be noted that the value "(a0 + a3) x D3 + a2 x D5 + a1 x D1", latched by the flipflop FF20, is not outputted as the decoded pixel data.

Clock Period CLK13

At the following clock period CLK13, as shown in Fig. 9, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK13. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F6, F2, F0, and F4 as the pixel data m1, m3, m5 and m6. respectively. These pixel data m1, m3, m5 and m6 are used for the calculation of the decoded pixel data f1 and f7, that is, the calculation of the second rows of the matrices in the equations (13) and (14). The output of the pixel data m1, m3, m5 and m6 allows the adders 23_1 to 23_3 to execute addition for calculating the decoded pixel data f1 and f7.

In the meantime, the multiplier 23_1

receives the pixel data F2(=a2) from the flipflop FF9, while the controller unit 30 sets the coefficient α to D5. This allows the multiplier 23_1 to calculate the product of a2 and D5 used in the equation (13). The product "a2 x D5" is stored into the flipflop FF14 at the end of the clock period CLK13.

Correspondingly, the multiplier 23_2 receives the pixel data F6(=a1) from the flipflop FF10, while the controller unit 30 sets the coefficient β to D1. This allows the multiplier 23_2 to calculate the product of a1 and D1 used in the equation (13). The product "a1 x D1" is stored into the flipflop FF15 at the end of the clock period CLK13.

Correspondingly, the multiplier 23_3 receives the sum "F0 + F4" (= a0 + a3) from the flipflop FF11, while the controller unit 30 sets the coefficient γ to D3. This allows the multiplier 23_3 to calculate the product of the sum "a0 + a3" and the coefficient D1 used in the equation (13). The product "(a0 + a3) x D1" is stored into the flipflop FF16 at the end of the clock period CLK13.

The output of the multiplier 23_4 is ignored.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the

flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to select the output of the flipflop FF19 by the controller unit 30. The fact that the flipflop FF17 output zero results in that the adder/subtractor unit 25 calculates the term "(a0 + a3) x D3 + a2 x D5 + a1 x D1", that is, the term "(f0 + f4) /2". The calculated term "(f0 + f4) /2" is latched by the flipflop FF19 at the end of the CLK13.

In addition, the decoded pixel data f0, which has been stored in the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK13. This allows the output of the decoded pixel data f0 from the flipflop FF20 at the following clock period CLK14.

Clock Periods CLK14 and CLK15

At the following clock period CLK14, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK14. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data F5, F1, F7, and F3 as the pixel data m1, m3, m5 and m7, respectively. These pixel data m1, m3, m5 and m7 and m6 are used for the calculation of the decoded pixel data f1 and f7, that is, the

calculation of the second rows of the matrices in the equations (13) and (14). The output of the pixel data m1, m3, m5 and m7 allows the adders 23_1 to 23_3 to execute addition for calculating the decoded pixel data f1 and f7. In addition, the multipliers 23_1 to 23_4 execute multiplication for calculating the decoded pixel data f1 and f7.

Furthermore, the adder/subtractor unit 25 calculates the sum of the outputs of the flipflops FF14 to FF17 and the selector 25_5. The selector 25_5 is controlled to invert the sign of the output of the flipflop FF19 by the controller unit 30. This results in that the adder/subtractor unit 25 calculates the decoded pixel data f4, because the adder/subtractor unit 25 calculates the difference when the value "(a0 + a3) x D3 + a2 x D5 + a1 x D1", which is the output of the flipflop FF19, is subtracted from the value "a5 x D6 + a7 x D4 - a6 x D2 - a4 x D0", which is the sum of the outputs of the flipflops FF14 to FF17. This calculation is equivalent to the calculation of the difference when the term "(f0 - f4) /2" is subtracted from the term "(f0 + f4) /2". The decoded pixel data f4 is latched by the flipflop FF19 at the end of the clock period CLK14.

Furthermore, the value "(a0 + a3) x D3 + a2

x D5 + a1 x D1", that is, the term "(f0 + f4) /2", which has been stored in the flipflop FF19, is transferred to the flipflop FF20 at the end of the clock period CLK14. It should be noted that the value "(a0 + a3) x D3 + a2 x D5 + a1 x D1", latched by the flipflop FF20, is not outputted as the decoded pixel data.

The decoded pixel data f4 is then transferred from the flipflop FF19 to the flipflop FF20 at the clock period CLK15. This allows the output of the decoded pixel data f4 from the flipflop FF20.

The same goes for the decoded pixel data f1 to f3, and f5 to f7.

Second Embodiment

In a second embodiment, the image processing apparatus is designed to perform the discrete wavelet transform using the reversible 5/3 filter in addition to the discrete wavelet transform using the irreversible 9/7 filter, and the discrete cosine transform.

Fig. 11 shows a block diagram of the image processing apparatus in the second embodiment. The image processing apparatus in the second embodiment is similar to that in the first embodiment, except for elements enclosed by a

dashed line 40. In detail, additional circuits (not shown) are disposed around the adder 21_1 and 21_2 to form a reversible processing circuit 41. Furthermore, selectors 42 and 45 are additionally disposed.

As shown in Fig. 14, the reversible processing circuit 41 includes selectors 46a and 46b, a shifter 47, a selector 48, and a complementer 49. The selector 46a selects one of the pixel data m1, the output of the flipflop FF14, and the output of the flipflop FF15 in response to a control signal from the controller unit 30. The output of the selector 46a is connected to the first input of the adder 21_1. The selector selects one of the pixel data m2 and the output of the flipflop FF15. The output of the selector 46b is connected to the second input of the adder 21_1. The input of the shifter 47 is connected to the output of the adder 21_1. The output of the shifter 47 is connected to the input of the flipflop FF9. The selector 48 selects one of the pixel data m3 and the output of the flipflop FF9. The output of the selector 48 is connected to the input of the complementer 49. The output of the complementer 49 is connected to the first input of the adder 21_2. The second input of the adder 21_2 receives the

pixel data m4.

Referring back to Fig. 11, the selector 42 selects one of the outputs of the multiplier 23_1, the flipflop FF9, and the flipflop FF15. The output of the selector 42 is connected to the flipflop FF14.

The selector 43 selects one of the outputs of the multiplier 23_2 and the flipflop FF10. The output of the selector 43 is connected to the flipflop FF14. The selectors 42 and 43 allow the image processor to disable the multipliers 23_1 and 23_2 during performing the discrete wavelet transform using the reversible 5/3 filter.

The selector 44 selects one of the outputs of the flipflops FF14 and FF10. The output of the selector 44 is connected to an input of the selector 45.

The selector 45 selects one of the outputs of the selector 45 and the limiter 26. The output of the selector 45 is connected to the input of the flipflop FF20. The selector 45 is controlled to select the output of the selector 44 during performing the discrete wavelet transform using the reversible 5/3 filter. This implies that the discrete wavelet transform using the reversible 5/3 filter does not require the adder 21_3, 21_4, the flipflops FF11 to FF13, the multipliers 23_3 to

23_5, the flipflops FF16 to FF18, the adder 25_1 to 25_4, the selector 25_5, and the flipflop FF19, and the limiter 26.

An explanation of the procedure of encoding through the discrete wavelet transform using the reversible 5/3 filter in this embodiment is given in the following.

As shown in Fig. 12, the encoding begins with the reception of the pixel data. The pixel data X(2n-2) to X(2n-8) are transferred to the flipflops FF0 to FF6, respectively, in synchronization with the clock signal. The pixel data X(2n-3) is associated with the pixel of interest, which is positioned in the odd numbered columns. The following is the explanation of the procedure of encoding the pixel data associated with the pixel of interest positioned in the odd numbered columns.

At a clock period CLK1, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK1. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data X(2n-2) and X(2n-4) as the pixel data m1, and m2, respectively.

The adder 21_1 calculates the sum of the pixel data X(2n-2) and X(2n-4), which is used in

the equation (5). The sum "X(2n-2) + X(2n-4)" is provided for the shifter 47.

The shifter 47 accomplishes 1-bit right-shift of the sum "X(2n-2) + X(2n-4)". This right-shift is equivalent to the division by 2, and thus the output of the shifter 47 is equal to [(X(2n-2) + X(2n-4))/2], where [x] is the floor function. The output of the shifter 47 is transferred to the flipflop FF9 at the end of the clock period CLK1. It should be noted that, in Fig. 12, numbers arranged in rows and columns denotes the indices specifying the pixels, the brackets "[ ]" represents that the data is processed by the floor function, and the symbols "*" represents that the associated data are intermediate results.

At the following clock period CLK2, pixel data X(2n-1) to X(2n-7) are transferred to the flipflops FF0 to FF6, respectively. The controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK2. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data X(2n-3) as the pixel data m3.

The complementer 49 develops a complement of the output of the flipflop FF9, that is, a

complement of [(X(2n-2) + X(2n-4))/2], and the developed complement is inputted to the adder 21_2. The adder 21_2 calculates the difference when [(X(2n-2) + X(2n-4))/2] received from the flipflop FF9 is subtracted from the pixel data m4. As described in the equation (5), this achieves the calculation of the encoded pixel data Y(2n-3). The encoded pixel data Y(2n-3) is transferred to the flipflop FF10 at the end of the clock period CLK2.

The encoded pixel data Y(2n-3) is transferred to the flipflop FF15 through the selector 43. The flipflop FF15 contains the encoded pixel data Y(2n-3) till the clock period CLK4 expires. The encoded pixel data Y(2n-3) is then transferred to the flipflop FF14 through the selector 42 at the end of the clock period CLK5. The flipflop FF14 contains the encoded pixel data Y(2n-3) till the clock period CLK6 expires. Then, the encoded pixel data Y(2n-3) is transferred to the flipflop FF20 through the selectors 44 and 45 at the end of the clock period CLK7. Finally, the encoded pixel data Y(2n-3) is outputted from the flipflop FF20 at the clock period CLK8.

Below is an explanation of the procedure of encoding the pixel data associated with the pixel of interest positioned in the even numbered

columns. The encoded pixel data for the even numbered columns is obtained using the intermediate results generated during the encoding for the odd numbered columns as described below.

At the clock period CLK3, the controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK3. In response to the control signal S1, the rearrange circuit 12 outputs the pixel data X(2n) and X(2n-2) as the pixel data m1, and m2, respectively.

The adder 21_1 calculates the sum of the pixel data X(2n) and X(2n-2), which is used in the equation (5). The sum "X(2n) + X(2n-2)" is provided for the shifter 47.

The shifter 47 accomplishes 1-bit right-shift of the sum "X(2n) + X(2n-2)". This right-shift is equivalent to the division by 2, and thus, the output of the shifter 47 is equal to [(X(2n) + X(2n-2))/2] in the equation (5). The output of the shifter 47 is transferred to the flipflop FF9 at the end of the clock period CLK3.

At the following clock period CLK4, pixel data X(2n+1) to X(2n-5) are transferred to the flipflops FF0 to FF6, respectively. The controller unit 30 develops the control signal S1

to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK4. In response to the control signal S1, the rearrangement circuit 12 outputs the pixel data X(2n-1) as the pixel data m4.

The complementer 49 develops a complement of the output of the flipflop FF9, that is, a complement of [(X(2n) + X(2n-2))/2], and the developed complement is inputted to the adder 21_2. The adder 21_2 calculates the difference when [(X(2n) + X(2n-2))/2] received from the flipflop FF9 is subtracted from the pixel data m4. As described in the equation (5), this achieves the calculation of the encoded pixel data Y(2n-1). The encoded pixel data Y(2n-1) is transferred to the flipflop FF10 at the end of the clock period CLK4. The encoded pixel data Y(2n-1) is transferred to the flipflop FF15 through the selector 43 at the end of the clock period CLK5.

At the following clock period CLK6, the adder 21_1 receives the pixel data Y(2n-1) from the flipflop FF15, the pixel data Y(2n-3) from the flipflop FF14. The adder 21_1 then calculates the sum of the pixel data Y(2n-1), Y(2n-3), and a constant of "2", that is, the term "Y(2n-1) + Y(2n+1) + 2" in the equation (6).

The shifter 47 accomplishes 2-bit right-

shift of the term "Y(2n-1) + Y(2n+1) +2". This 2-bit right-shift is equivalent to the division by 4, and thus, the output of the shifter 47 is equal to [(Y(2n-1) + Y(2n+1))/2] in the equation (5). The output of the shifter 47 is transferred to the flipflop FF9 at the end of the clock period CLK6.

At the following clock period CLK7, pixel data X(2n+4) to X(2n-2) are transferred to the flipflops FF0 to FF6, respectively. The controller unit 30 develops the control signal S1 to indicate the rearrange circuit 12 to execute the procedure defined for the clock period CLK7. In response to the control signal S1, the rearrangement circuit 12 outputs the pixel data X(2n-2) as the pixel data m4.

The adder 21_2 calculates the sum of the pixel data X(2n-2) and the output of the flipflop FF9, that is, [(Y(2n-1) + Y(2n+1))/2]. This achieves the calculation of the right hand of the equation (6), that is, the encoded pixel data Y(2n-2). The encoded pixel data Y(2n-2) is transferred to the flipflop FF10 at the end of the clock period CLK7. The pixel data Y(2n-2) is then transferred to the flipflop FF20 through the selectors 44 and 45 at the end of the clock period CLK8. This allows the output of the

encoded pixel data Y(2n-2) from the flipflop FF20 at the following clock period CLK9.

Fig. 13 is a timing chart describing the procedure of decoding through the discrete wavelet transform using the reversible 5/3 filter. The procedure of the decoding is almost similar to the aforementioned encoding except for that the equations (7) and (8) are used in place of the equations (5) and (6). Therefore, detailed explanation is not given.

As thus-described, the image processing apparatus in accordance with the present invention can perform both the discrete wavelet transform and the discrete cosine transform by using the same circuitry. This effectively reduces the necessary hardware resources.

Although the invention has been described in its preferred form with a certain degree of particularity, it is understood that the present disclosure of the preferred form has been changed in the details of construction and the combination and arrangement of parts may be resorted to without departing from the scope of the invention as hereinafter claimed.